Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Modify Bidirectional Associative Memory (MBAM)

Nisreen Abd Alhadi Jabr[1], Emad I Abdul Kareem[2]
[1,2]College of education in Computer since, AL Mustansiriyah University, Iraq. Baghdad

Abstract— Associative memory is a data collectively stored in the form of a memory or weight matrix, which is used to generate output that corresponds to a given input, can be either auto-associative or hetero-associative memory. A Bidirectional Associative memory neural network is one of the most commonly used neural network models for hetero-association and optimization tasks, it has several limitations. For example, it is well known that Bidirectional Associative memory neural networks has limited stored patterns, local minimum problems, limited noise ratio and shifting and scaling problems. This research will suggested to improve the Bidirectional Associative Memory neural network by modifying the net architecture, learning and convergence processes, and this modification is to increase the performance of associative memory neural network by avoiding most of the Bidirectional Associative Memory neural network (BAM) limitations. This research will proposed modify Bidirectional associative memory (MBAM) to improve the efficiency of BAM in order to decrease the network size and weight size. In additional to increase the ability for noise robust as well as speed up its learning and convergence process. The results proved that the MBAM net can learn and recognize unlimited patterns in varying sizes with the acceptable percentage noise.

Key Words: Neural network, Associative memory, Bidirectional Associative memory neural network, Pattern recognition

I. INTRODUCTION

Bidirectional Associative Memory neural network is one of the neural networks that are used for hetero-association and optimization tasks. This net has many limitations that are effected its performance. These limitations consisted: firstly the number of patterns that can be stored and accurately recalled is severely limited. Secondly it is obviously desirable to reach a global minimum rather than setting down at a local minimum as it can happen while using the energy function. Thirdly correlation problem happens when an input pattern share many bits with another pattern. Fourthly the ratio of missing and mistake data in the input patterns is limited. Fifthly, it is impossible to retrieve the stored pattern when it enters to the network with shifting or scaling.

MBAM net avoids most of the BAM limitation, except two which it the shifting and scaling problem, in addition to the smaller size of net and the efficient learning and convergence process.

This research proposed a present to improve efficiency of MBAM in order to decrease the network size and weight size. In additional to increase the ability for noise robust as well as speed up its learning and convergence process. Similar to the BAM neural network and MBAM is a two layer neural network, which uses hetero-association tasks and work in two phases (learning and convergence phases).

The experiments performed show promising results when MBAM shows high efficiency to recognize many noisy patterns in varying size comparing with the traditional Bidirectional Associative Memory (BAM) neural network.

II. BIDIRECTIONAL ASSOCIATIVE MEMORY (BAM) NEURAL NETWORK MODIFICATIONS

Other Bidirectional Associative Memory (BAM) neural network modifications have been made to improve efficiency. All of these have used the amendments for a specific purpose, thus they are away from my work, and the details are as follows. Christophe Tremblay and et al. proposed an adjustment on BAM network that increases its performance in a rapid learning condition while processing memory capacity is limited. Results show that the modification to the original learning rule of the BHM (Bidirectional Hetero-associative Memory) leads to improved performance when rapid learning is required. Moreover, the model preserves its high memory load capacity in a standard learning [1]. Hoa Thi Nog and the Duy Bui. Studied learning strategy pair weights in the interactive learning algorithm are modified therefore a learning process is faster and the ability of recall is larger or equal to than other BAMs [2].

BAM models have difficulties learning nonlinear separable tasks, the problem comes from the fact that these networks do not have a hidden layer or a highly nonlinear transmission function. Christophe Tremblay and et al. study introduces a modification of the architecture of a given type of BAM by adding an unsupervised pathway to the original BAM structure. Results showed that the modification allows the network to perform nonlinearly separable associations such as the n-bit parity task and the double-moon problem. The network is able to associate more difficult types of problems while keeping the same learning and the transmission function. This study could lead to enhanced cognitive models capable of modeling a wider set of associations [3].

Mo Wei and et al. is introduced a strategy named Bit Importance Strategy (BIS) to improve the performance of Bidirectional Associative Memory (BAM) in cases in which differences in importance among bits must be considered. The main advantage of this new strategy lies in the improvement of the recall success ratio. Simulation results are given, with an application to a simplified version of a military battlefield operation, based on a theoretical analysis [4].

Osana et al. is proposed Chaotic Bidirectional Associative Memory (CBAM), It has a very simple structure because it merely uses chaotic neurons in a part of the conventional Bidirectional Associative Memory. It can deal with one-to-many associations by memorizing each training pair with its own contextual information and using chaotic neurons corresponding to the contextual information. It has a strong possibility that all desired patterns can be recalled in much shorter time than a method using noise. In the CBAM, each training pair is memorized together with its own contextual information. Since the chaotic neurons used in a part of the network corresponding to the contextual information change their states by chaos, the CBAM can recall plural desired patterns dynamically [5].

Yano and Osana. is proposed the Chaotic Complex value BAM (CCBAM) which is based on CBAM. Both CBAM and CCBAM have the same structure, but the neurons in the second memory follow the model of chaotic neuron introduced by Aihara et al. [1990]. These neurons are complex-valued and they can realize dynamic associations with multivalued patterns. Also, CCBAM can realize one-to-many associations [6]. Sylvain Chartier and Mounir Boukadoum. is presented the new BAM model that uses a chaotic output function operating in chaos mode during recall. Results show that the model developed well-defined, with the result the chaotic BAM is more tolerant to noise than a regular fixed point BAM. This is concluded from simulations that showed the superior performance of the new model when compared to the original BAM architecture or when using a linear technique such as the pseudo-inverse [7].

Most former models been able to recall all the trained patterns, Maria Elena and et al. is introduced a new model of bidirectional associative memory which is not iterative and has no stability problems. It is based on the Alpha-Beta associative memories. This model allows, besides correct recall of noisy patterns, perfect recall of all trained patterns, with no ambiguity and no conditions. An example of fingerprint recognition is presented [8].

TAE-Dok Eom and et al. studied the hamming distance in recall procedure of usual asymmetrical BAM is replaced with modified Hamming distance by introducing a weighting matrix into connection matrix. This generalization is validated to increase storage capacity, to lessen spurious memories, and to enhance noise immunity using simulation work. Among many efforts to improve the performance of Kosko BAM (KBAM) by introducing new learning algorithms and adding dummy neurons, more layers, or interconnections inside each layer, the symmetrical BAM using the Hamming stability learning algorithm (SBAM) achieves the highest performance. The general BAM (GBAM) used linear separability condition and increased the capacity slightly greater than the number of neurons in its layer. Generalized asymmetrical BAM (GABAM), simply multiplying input weighting matrix to the transition matrix of KBAM and deriving unique learning algorithms outperform the previous model [9].

III. ASSOCIATIVE MEMORY

It is believed that human memory is stored in the form of complex interconnections among various neurons. In artificial neural networks simulating associative memory, data are collectively stored in the form of a memory or weight matrix, which is used to generate output that corresponds to a given input. Such a process is referred to as learning or storing the desired patterns, while the retrieval or recall process is referred to as the generation of an output pattern [10]. Figure 1 shows a general block diagram of an associative memory system performing an associative mapping of an input vector x into an output vector v (see equation 1).

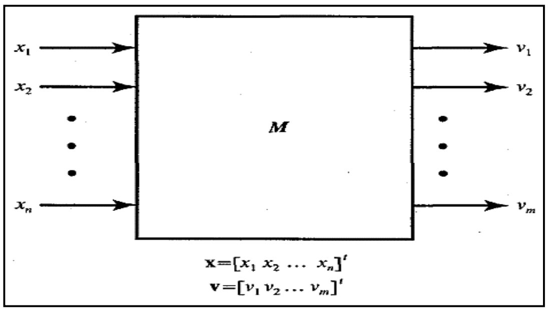

Figure 1. Block diagram of an associative memory.

$$v = M[x] \qquad (1)$$

The system shows map vectors x to vectors v, in the pattern space and output space, respectively, by transformation. The operator M denotes a general nonlinear matrix-type operator; which has different meanings for each of the memory models. Its form, in fact, defines a specific model that needs to be carefully outlined for each type of memory, whereas its structure reflects a specific neural memory paradigm. For dynamic memories, M also involves a time variable. Thus, v will be available at the memory output at a later time when the input is applied [11].

An associative memory can be applied in either auto-associative or hetero-associative applications. Mathematically, it is a mapping from an input space to an output space. In other words, when the network is presented with a pattern similar to the member of the stored set, it may associate the input with the closed stored pattern [12].

Generally, in hetero-associative applications, the dimensions of the input space and the output space are different, as illustrated in Figure 2. Nevertheless, the training input and target output vectors of auto-associative memory are identical, as illustrated in Figure 3 [12] [11].

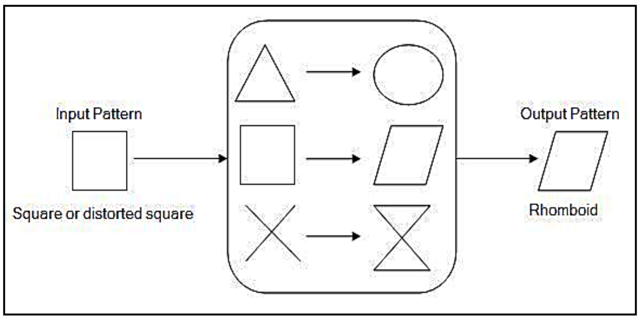

Figure 2: Hetero-association response

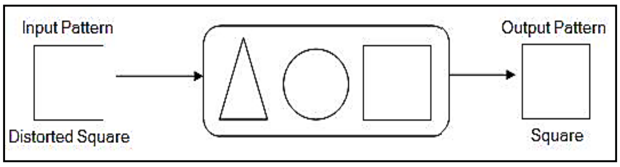

Figure 3: Auto-association response.

One of the hetero-associative memory neural networks is the Bidirectional Associative Memory (BAM), which is presented in the next section.

3.1 Bidirectional Associative Memory (BAM) Neural Network

This section presents one of the neural network models; i.e., the "Bidirectional Associative Memory (BAM) network", which is the most commonly used for hetero-association and optimization tasks. The Bidirectional associative memory is hetero-associative, developed by Kosko (1988, 1992a). A BAM consists of neurons arranged in two layers X and Y. The neurons in one layer are fully interconnected to the neurons in the second layer. There is no interconnection among neurons in the same layer. The weight from layer X to layer Y is same as the weights from layer Y to layer X.

Dynamics involves two layers of interaction. Because the memory process information in time and involves bidirectional data flow, it differs in principle from a linear association, although both networks are used to store association pairs. It also differs from the recurrent auto- associative memory in its update mode [11]. The next subsection will discuss the architecture of the Bidirectional associative Memory neural network.

3.1.1 Bidirectional Associative Memory (BAM) Neural Network Architecture

The bidirectional associative memory Neural Network is hetero-associative system. The single-layer nonlinear feedback BAM network (with hetero-associative content- addressable memory) has n units in its X-layer and m units in its Y-layer. The connections between the layers are bidirectional; i.e., if the weight matrix for signals sent from the X-layer to the Y-layer is W, the weight matrix for signals sent from the Y-layer to the X-layer is WT, show Figure 4, Architecture of the BAM [12].

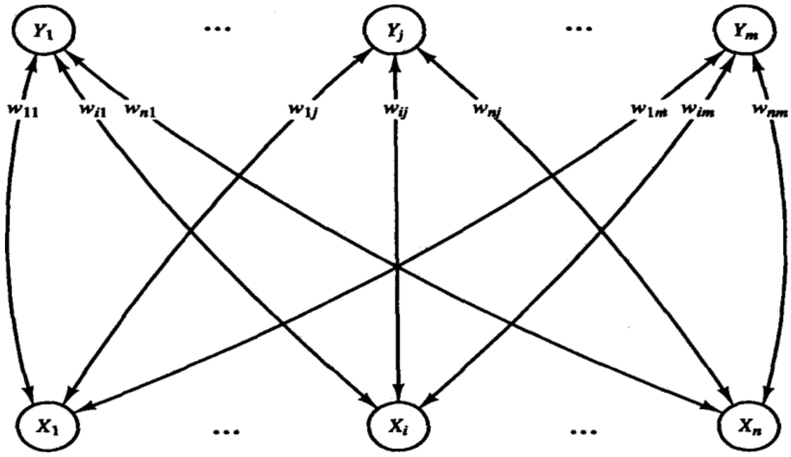

Figure 4: The architecture of the Bidirectional Associative Memory (BAM), where: Wij: Weights Y1, Y2,…, Yn and X1, X2,…, Xn Neurons.

3.1.2 Bidirectional Associative Memory (BAM) Neural Network Algorithm

The Bidirectional associative memory (BAM) neural network works in two phases, which are the learning phase and convergence phase, as shown in Figure 5. In the algorithm, the learning phase will create weight matrix W *(wij*) for the training patterns, which will not change afterwards. If there is more than one training pattern (S1 (p), S2 (p), …, Sn(p)), and target vectors(t1(p), t2(p), …, tm(p)) [12].

After initializing the Bidirectional associative memory (BAM) network with an unknown input pattern, the convergence phase will start such an operation and will be repeated until there is no change in the BAM network output throughout successive iterations. Then, the process will be stopped and the recovered pattern will be matched with the stored patterns and assigned a class [13][14].

To prove that the previous mentioned iterated convergence process are stable one should consider the value of the energy function, which plays a vital role in the authenticity of the process as holes [15].

This function decreases as the system states change. Such a function needs to be found and watched as the network operation continues from one cycle to another. In the BAM network, an energy function is a function that is bounded below and whenever the state of any unit changes. I.e., this function always decreases gradually to reach a minimum and then stops when the network is stable. As stated below [12]:

$$L = -0.5(xWy^T + yW^Tx^T) \qquad (2)$$

Where

L = an artificial network energy.

Algorithm 1: Bidirectional Associative Memory Neural Network (BAM).
Step 0. Initialize the weights to store a set of P vectors; initialize all activations to 0. **Step 1.** For each testing input, do Steps 2-6. **Step 2a.** Present input pattern x to the X-layer. (I.e., set activations of the X - layer to current input pattern). **Step 2b.** Present input pattern y to the Y-layer. (Either of the input patterns may be the zero vector.) **Step 3.** While activations are not converged, do Steps 4-6. **Step 4.** Update activations of units in Y-layer. Compute net inputs: $y\_in_j = \sum_i w_{ij} x_i.$ Compute activations: $y_j = f(y\_in_j).$ Send signal to X-layer. **Step 5.** Update activations of units in X-layer. Compute net inputs: $x\_in_i = \sum_j w_{ij} y_j.$ Compute activations: $x_i = f(x\_in_i).$ Send signal to Y-layer. **Step 6.** Test for convergence: If the activation vectors x and y have reached equilibrium, then stop; otherwise, continue.

Figure 5. Algorithm 1: Bidirectional Associative Memory neural network Neural Network [12]

3.1.3 Bidirectional Associative Memory (BAM) neural network limitations

Experimentally, the number of binary patterns that can be stored and recalled in a Bidirectional associative memory (BAM) neural net with reasonable accuracy are limited. Fausett [12] performed a detailed theoretical analysis of the information capacity of a Bidirectional associative memory (BAM) neural net and the results are given approximately by Equation 3 The net may converge to a novel spurious pattern different from all exemplar (or training) patterns.

$$p \approx 0.15n \qquad (3)$$

During the network convergence process, it is obviously desirable to reach a global minimum rather than settling down at a local minimum, as can happen while using the energy function. Figure 6 clarifies the distinction between a local minimum and a global minimum.

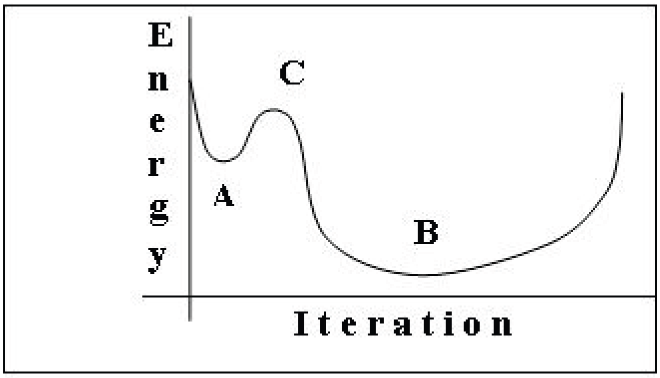

Figure 6. Local and global minima.

In this figure we find the graph of an energy function and the two points: point A and point B. These points show that the energy level at that point is smaller than the energy levels at any point in their vicinity. Thus we can say they represent points of minimum energy, be it the overall or global minimum. As you can see, point B is where the energy level is smaller than at point A. Thus point A corresponds only to a local minimum. However, it is desirable to get to point B in the pursuit of the minimum of the energy function. If point C is reached, further movement toward point B and not point A is desired. Similarly, if a point near point A is reached, the subsequent movement should avoid settling but instead should proceed to point B [14].

On the other hand, Fausett [12] mentioned the stability problem that occurs because of the correlation problem. The correlation problem happens when one or more patterns have similar elements (or bits). A pattern is considered unstable if it is applied and the net converges to some other pattern. Therefore, with the Bidirectional associative memory (BAM) neural network all the stored patterns must be orthogonal to each other.

The allowable percentage noise pattern (missing and mistake pattern elements value) to be recognized is limited [13]. There is the possibility of converging one of the stored patterns in reversed values (all 1s become 0s or -1s using bipolar values and vice versa) [12]. Finally, this network is incapable of converging towards any stored pattern when it enters the network after a shifting or/and scaling operation [13].

IV. MODIFY BIDIRECTIONAL ASSOCIATIVE MEMORY (MBAM).

The size of a Bidirectional associative memory (BAM) network (number of neurons in the network) is dependent on the pattern length used by the network (i.e., pattern with length ten requires a BAM network of size ten). Since MBAM is a modified BAM neural network, MBAM obeys the same size property. According to the first principle, the pattern will be divided into a number of vectors with length two; this means that the size of the network will be fixed (i.e., two). Therefore, the network deals with parts of the pattern instead of the entire pattern as one vector. This leads to the advantage of working with the smallest network size regardless of the pattern length, as well as multiple connections between the two neurons. The new architecture permit the possibility of avoiding learning the same vectors (which represents a certain part of the pattern) several times. This arrangement achieves the second principle described above.

With a bipolar pattern representation, the elements will be either 1 or -1. The reason for choosing this length of vector is the shortest even length of any vector is two. However, just as in the traditional BAM neural network, each node is connected to every other node but not to itself. These connections represent the corresponding weight of each vector in the pattern. Although the expected number of vectors is, the number of connections will be just four, an advantage that helps deal with the smallest network size regardless of the pattern length. Technically, as with traditional BAM nets, this adapted net has two phases (learning and convergence phases). In this research, these two processes will be modified.

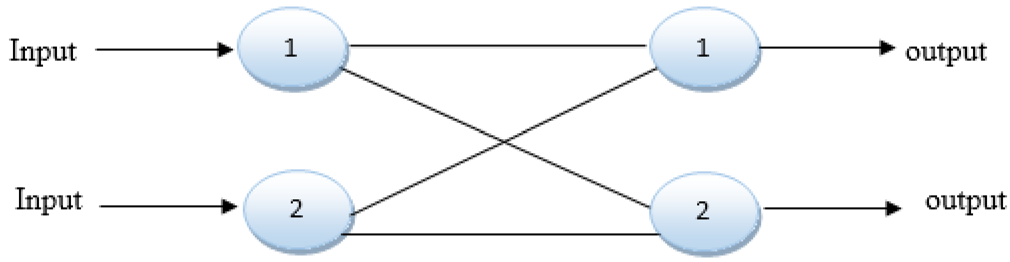

Figure 7: The Modified bidirectional associative memory (MBAM)

After presented the proposed new structure, it became important to provide algorithms that are approved this new structure for both learning and convergence phase.

4.1 Learning Phase (learning process)

This section presents the learning algorithm shown in Figure 8. The network is able to recognize all the required bipolar patterns through the implementation of this phase. Therefore, the input for this phase is a flow of patterns and codes to be considered as training patterns for this network. The output of this phase is the results of the learning process, which is stored in a specific lookup table. Since the training patterns are divided into vectors with length two, these results are a set of weights indexes for each vector of the training patterns, which are called stored vector weights *svw*.

Algorithm 2: Learning phase for MBAM
Input: training patterns *p* with code *c*.
Output: lookup table for all *n* corresponding stored patterns.
Step1: Repeat steps 1.1 and 1.2 to the end of training pattern *p* with code *c*:
Step 1.1: Divide the training pattern *p* to *n* vectors *v* with length two.
Step 1.2: For each vector *v*, repeat steps 1.2.1, 1.2.2 and 1.2.3:
Step 1.2.1: Assign the weight for each *n* vectors *v*, weights matrix *sw* as follows: $svw_i = v_i * c$
Where: *c* is code
Step 1.2.2: Assign the stored vector's weight svw as follow: $svw = f(Dcode(v)) \begin{cases} 0 & \{ means w0 \\ 1 & \{ means w1 \\ 2 & \{ means w2 \\ 3 & \{ means w3 \end{cases}$
Where: Decode is a function to convert the binary number to decimal number.
Step 1.2.3: Save svw for this vector in the lookup table.
Step 2: End.

Figure 8. The algorithm 2 of the learning phase.

4.2 Convergence Phase (convergence process)

The output of this phase is dependent on the learning phase. Although the modification has been carried out during the learning phase, as it is necessary to have a comprehensive modification for the convergence phase (see Figure 9 and Figure 10) in order to ensure the overall efficiency of the Modified Bidirectional

Associative Memory (MBAM). Convergence phase have two Bi-direction, first direction is code to pattern and second direction is pattern to code.

4.2. a. Convergence code Phase

Algorithm 2: a. Convergence code Phase for MBAM
Input: n of unknown patterns p. Output: Convergence code c. Step 1: Repeat steps 1.1 and 1.2 until the unknown pattern p is ended: Step 1.1: Divide the unknown pattern p to n vectors v with length two. Step 1.2: Sum up the energy function for all n vectors v in the unknown pattern each with its corresponding vector in the stored patterns: $$ep = -0.5 \sum_{i=1}^{n} v_i * svw_i * y_i$$ Where y_i is : $$y_i = v_i * svw_i$$ Step 2: Determine the stored code number $minc$ with the minimum energy function to converge the unknown pattern towards it: $$minc = \min(ep)$$ Where the min function is to determine the minimum energy function in ep array. Step 3: End.

Figure 9: Algorithm 2: a. The convergence code algorithm for MBAM.

4.2. b. Convergence pattern phase:

Algorithm 2: b. Convergence pattern phase for MBAM
Input: n of unknown codes c. Output: Convergence pattern *CP*. Step 1: Repeat steps 1.1 until the lookup table is ended: Step 1.1: Find energy function ep: $$ep = -0.5 * \sum x_i . c$$ Where x_i : $$x_i = y_i * svw_i$$ Where y_i : $$y_i = c * svw_i$$ Where c: is unknown code. svw: is matrix weights stored in lookup table. Step 2: Determine the stored pattern number $minp$ with the minimum energy function to converge the unknown code towards it: $$minp = \min(ep)$$ Where the min function is to determine the minimum energy function in ep array. Step 3: Build the final converge pattern cp: $$cp = svw_{minp} * c$$ Step 4: End.

Figure 10: Algorithm 2: b. The convergence pattern algorithm for MBAM.

V. MBAM COMPLEXITY

A useful measure of the execution time or memory to segment of an algorithm is Big-O [16]. Figure 11 shows how the MBAM algorithm work.

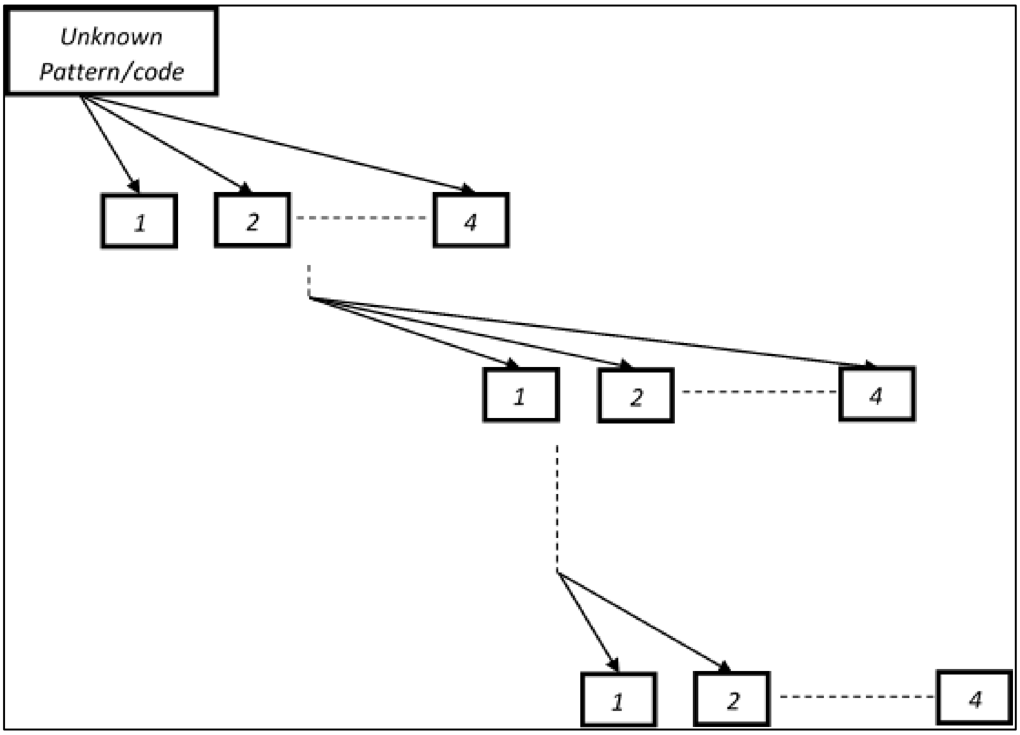

Figure 11: MBAM convergence of the unknown pattern/code towards one of the stored patterns.

The energy function of each level between the stored vector and the unknown patterns/code will be calculated. Such a calculation will be based on both the value of the energy function until the current level and the energy function in this level. According to the value of the energy function, MBAM will determine the convergence direction in the next level. The maximum number of branching is 4 because there are only 4 possible vectors, the maximum path length for a pattern with n elements is k which is equals to $n/2$, ($n/2$ is equals to the number of vectors with two elements on a pattern with n elements). Therefore, the Big-O of this process is O (4× $(n/2)$). On the other hand, the Big-O of the Bidirectional associative memory neural network (BAM) is $O(n^2)$ owing to the weight matrix that is created by multiplying the pattern with itself. Accordingly, it is clear that the complexity of the MBAM is lower than that of the Bidirectional associative memory neural network (BAM) owing to n and n^2 order for MBAM and Bidirectional associative memory neural network (BAM) respectively.

VI. EVALUATION OF MBAM

This section presents three experiments carried out to evaluate the MBAM associative memory. The experimental protocols were applied to both the traditional BAM neural network and the MBAM associative memory to allow comparison. The experiments considered the efficiency of the MBAM associative memory and were compared with the traditional BAM neural network by analyzing the noise rate (missing and mistake bits), a large number of training patterns and the small size of the net, which was two with all training pattern lengths and/or numbers.

6.1 Different Number of Stored Patterns vs. Convergence Rate

Volume 02, Issue 08, [August– 2015] ISSN (Online):2349–9745 ; ISSN (Print):2393-8161

In this experiment, for both MBAM associative memory and the traditional BAM network, the task was to learn the maximum number of patterns with codes for pattern; the process was stopped when one/both of them completely failed to recognize the stored patterns. The test patterns that were presented were alphabetical letters in 32×32 size without noise (see Figure 12).

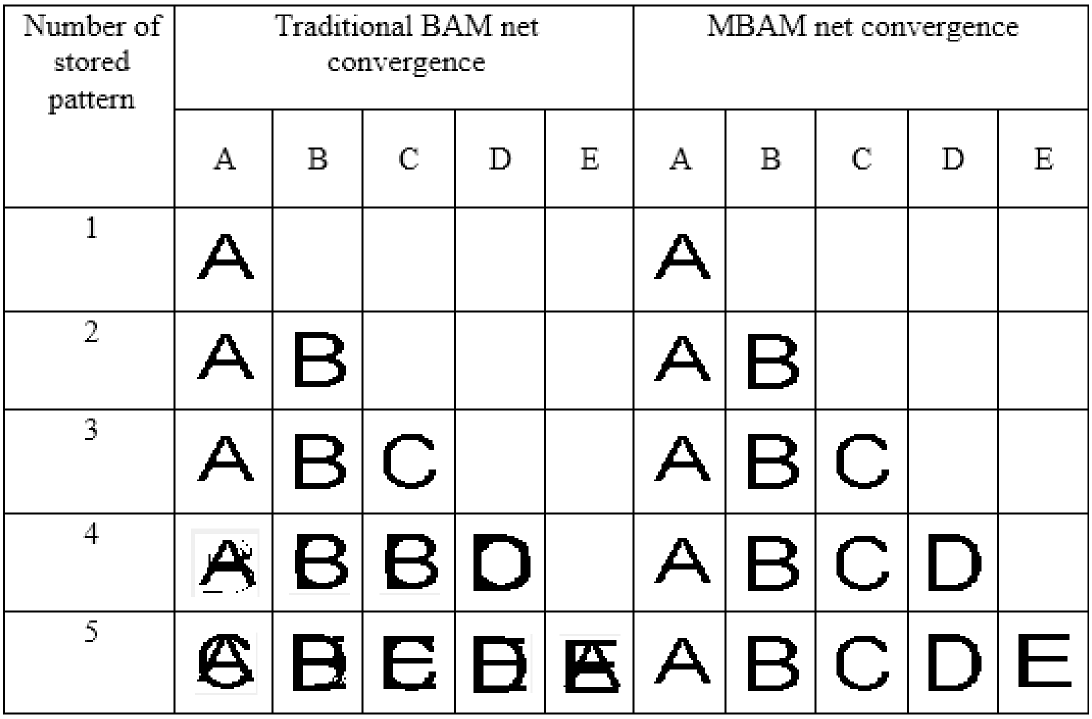

Figure 12. Traditional BAM net and MBAM learned four alphabetical English letters (A, B, C, D and E) with the converge patterns for both of them.

In Figure 12, it is clear that the learning process happened gradually, starting from pattern one to five, to make sure that both were able to retrieve most of the patterns until one of them or both reached the case of complete failure. The BAM neural network began fail to recognize some patterns when the number of patterns that had been learned was 4. Complete failure occurred when the number of patterns became 5.

6.2 Results Discussion and Analysis

Due to a correlation problem, the convergence rate of the traditional BAM net decreased when the number of stored patterns increased. Convergence failure started with 4 stored patterns and complete failure to recognize any stored pattern occurred with 5 stored patterns. Whereas MBAM associative memory kept the same convergence rate (which was 100%) even when the number of stored patterns increased to 5 (see Figure 13).

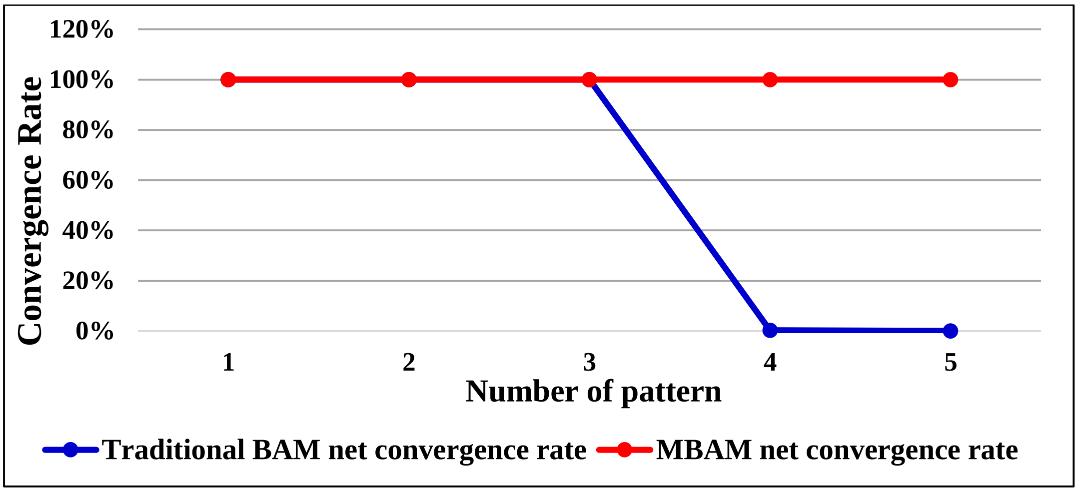

Figure 13. Diagram illustrate a comparison between traditional BAM net and MBAM net for number of stored patterns vs. Convergence rate for both of them.

The grid size of the stored patterns in this experiment was 32×32. Thus, the traditional BAM net size was 1024 nodes (or neurons), however, the MBAM associative memory network was maintained at a 2 nodes size because, as mentioned previously, the traditional BAM net size depends on the pattern size, whereas the MBAM net size is 2 nodes with any pattern size.

6.3 Different Noise Rates vs. Convergence Rate Depending on the Previous

Based on the previous experiment, traditional BAM net convergence maintained efficiency at 100% with input patterns having 0% noise until the number of stored patterns was 3 (see Figure 11). Thus, in the experiment, the MBAM associative memory and traditional BAM nets learned just 3 patterns with a grid size of 32×32. This experiment had three tests with each having three different random noise patterns for each pattern (A, B and C). The convergence process was implemented with both by testing the input of these patterns with 10% to 90% random noise.

Table I show the responses of both networks when the same three forms that the networks had learned previously were presented with different noise ratios ranging from 10% to 90%.

Noise Ratio	BAM	MBAM
	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 32×32
10%	96.67%	100%
20%	88	100%
30%	86.33%	100%
40%	79.67%	99.67%
50%	75.33%	96.33%
60%	70.67%	82.33%
70%	65.67%	64.67%
80%	59.67%	48.33%
90%	57.67%	36.67%

Table I: illustrating the comparison between the traditional BAM net, MBAM net for input patterns with different noise rates vs. convergence rates for each.

6.4 Results Discussion and Analysis

The efficiency of the traditional BAM network convergence was 10% the noise rate is 96.67%. At this noise rate, the network failed to converge some of the stored. Compared to the MBAM associative memory, the convergence efficiency was high (100%) until the noise rate was 40%.

It was also observed that the traditional BAM net sometimes failed to converge any stored patterns, compared with MBAM associative memory that converged, even though with low convergence rates. In the case of net failure, the traditional BAM net converged with patterns that were not related to any stored patterns, where the MBAM associative memory net only converged incorrectly to one of the stored patterns. Finally, Figure 14 illustrates a complete comparison between results for both networks.

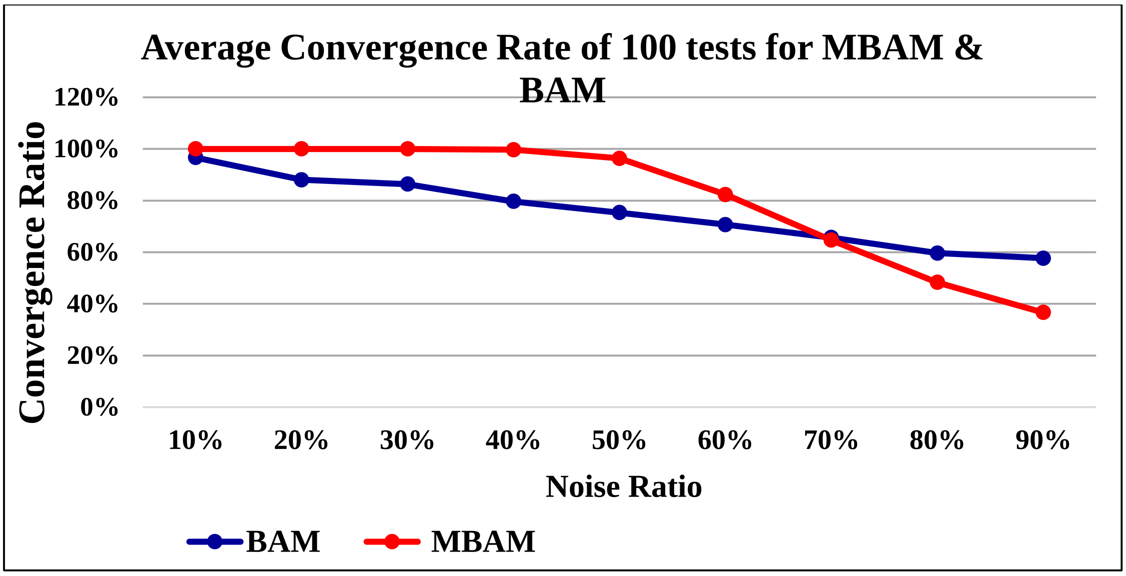

Figure 14. Diagram illustrating the comparison between the traditional BAM net and the MBAM associative memory network for input patterns with different noise rates vs. convergence rates for each.

The MBAM associative memory efficiency was high (100%) until the noise rate became 40%. Since the MBAM associative memory processed the patterns as vectors with size two, with small patterns there was the possibility for the random noise operation to change at least one element (bit) in all these vectors. Therefore, the MBAM associative memory may not have recognized the pattern because its convergence algorithm depended on these vectors. Exclusively, in this case, the MBAM associative memory could not recognize these patterns properly, leading to an incorrect convergence response. Experimentally, the potential occurrence of this case was reduced whenever the pattern size was increased because the effect of the random noise operation was reduced. This means, with or without random noise, MBAM associative memory is efficient with a large pattern.

6.5 Different Patterns Sizes with Different Noise Rates vs. Convergence Rates for MBAM Associative Memory

In this experiment, 100 patterns were presented in various sizes to the MBAM associative memory network with different noise ratios as unknown images to determine convergence rates (see Table II).

Noising Ratio	Convergence ratio of Pattern with size 10×10	Convergence ratio of Patterns with size 16×16	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 64×64	Convergence ratio of Patterns with size 128×128
10%	100%	100%	100%	100%	100%
20%	99.67%	100%	100%	100%	100%
30%	99%	99.67%	100%	100%	100%
40%	93%	95%	99.67%	100%	100%
50%	80.33%	84.33%	96.33%	100%	100%
60%	61.33%	71%	82.33%	98 %	100%
70%	48.33%	55.33%	64.67%	77.33%	100%
80%	43.33%	45.33%	48.33%	54.67%	100%
90%	39.33%	40%	36.67%	38%	100%

Table II. Five tests of 100 different size patterns for the MBAM associative memory net with convergence rates

6.6 Results Discussion and Analysis

Table II shows that the MBAM associative memory convergence rate was more efficient when the size of the patterns was increased (see Figure 15). Once again this situation occurred due to the random noise operation, which was carried out earlier in the Second Experiment.

Figure 15 Diagram illustrating a comparison between the five tests of 100 different size patterns with different noise rates for the MBAM net vs. convergence rates for each pattern size.

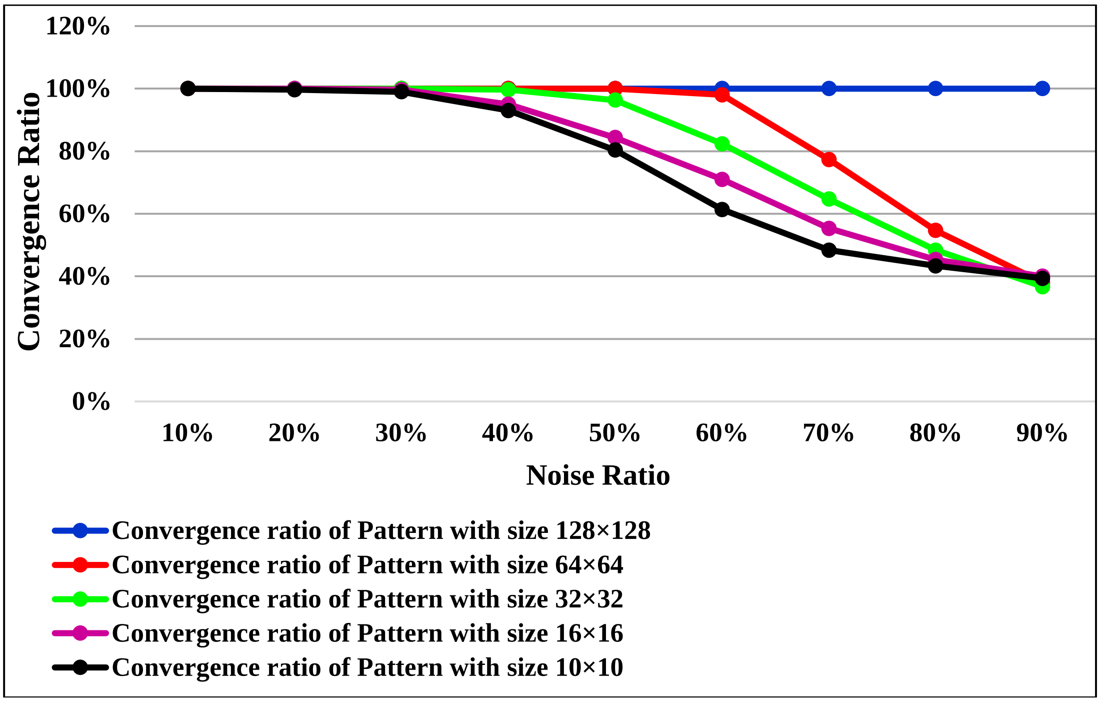

Figure 15: Diagram illustrating a comparison between the five tests of 100 different size patterns with different noise rates for the MBAM net vs. convergence rates for each pattern size.

VII. CONCLUSION

To enhance the efficiency of the association tasks, this paper proposed MBAM as a modified BAM neural network via modified the net structure in addition to the learning and convergence process. For structure modification the size of the net becomes fixed (two neurons) with any patterns size. This size of net caused that the size of the learning weigh matrix becomes small (four matrices). For learning process modification this net reached unlimited stored pattern, i.e. MBAM still efficient even when the number of stored patterns increases. Finally, for convergence process correlation problem has been solved, thus with MBAM net, it can store and retrieve the correlation patterns efficiently. In the other hand MBAM net still failed to recognize any stored patterns if scaling or shifting version of this stored patterns are presented to the net.

REFERENCES

1. Christophe, T., et al., Bidirectional Associative Memory for Short-term Memory Learning. The Annual Meeting of the Cognitive Science Society COGSCI, 2014.
2. Nong, H.T., and Bui, T. D., A fast iterative learning strategy for Bidirectional Associative Memory. In Proceedings of the Conference on Information Science and Technology, 2012.
3. Christophe, T., et al., Bidirectional Associative Memory and Learning of Nonlinearly Separable Tasks. ICCM International Conference on Cognitive Modelling, 2013.
4. Mo Wei, et al., Bit Importance strategy for Bidirectional Associative Memory. IEEE Conference on Decision and Control, and the European control conference, 2005.
5. Osana, Y., et al., Chaotic bidirectional associative memory. In Proceedings of the IEEE International Conference on Neural Networks, 1996. Vol. 2, p. 816-821.
6. Yano, Y., and Osana, Y., Chaotic complex-valued bidirectional associative memory. In Proceedings of the International Joint Conference on Neural Networks, 2009.
7. Sylvain, C., and Mounir, B., A Chaotic Bidirectional Associative Memory. In Springer International Conference, 2011.
8. Maria, E., et al., A New Model of BAM: Alpha-Beta Bidirectional Associative Memories. In Lecture Notes in Computer Science, 2006.

9. Tae-Dok, E., et al., Generalized Asymmetrical Bidirectional Associative Memory. Applied Mathematics and Computation, 1999. Vol. 127, p.221-233.
10. Kishan Mehrotra, C.K.M.a.S.R., Elements of Artificial Neural Networks. Bradford Books, Complex Adaptive Systems, New York, USA, 1996.
11. Zurada, J.M., Introduction to Artificial Neural System. PWS Publishing Co., Boston, MA, USA, 1999.
12. Fausett, L., Fundamental of Neural Networks Architecture, Algorithms and Application Prentice Hall, 1994.
13. Lippmann, R.P., An Introduction to Computing with Neural Nets. IEEE Assp Magazine 1987.
14. V, S.G., Neural Network and its Application in Pattern Recognition. Institute of Technology, Bombay, India, 2004.
15. Rao, V.B., C++ Neural Networks and fuzzy logic. MT Books, IDG Books Worldwide, 1995.
16. Cormen Thomas H., Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. (2001) *Introduction to Algorithms (*2nd ed.). McGraw-Hill Higher Education.